Journal of Autonomous Intelligence (2024) Volume 7 Issue 4
doi: 10.32629/jai.v7i4.1455

ORIGINAL RESEARCH ARTICLE

Word translation for Indo-Aryan languages using different retrieval techniques

Kiranjeet Kaur[1,2,*], Shweta Chauhan[1,2,3]

[1] *Department of Computer Science and Engineering, Chandigarh University, Mohali, Punjab 140413, India*

[2] *University Centre for Research & Development, Chandigarh University, Mohali, Punjab 140413, India*

[3] *Apex Institute of Technology, Chandigarh University, Mohali, Punjab 140413, India*

*** Corresponding author:** Kiranjeet Kaur, kiranjeet.cse@gmail.com

ABSTRACT

The study of Natural Language Processing has been revolutionized by word embedding, enabling advanced language models to understand and generate human-like text. In this research article, we delve deep into the world of word embedding, aiming to provide a comprehensive exploration of its underlying principles, methodologies, and applications. One important factor that affects many multilingual language processing activities is the word translation or incorporation of bilingual dictionaries. We used bilingual dictionaries or parallel data for translation from one language to another. For this research work, this problem is addressed, and also generating the best cross-lingual word embedding for the different language pairs. So, we are using an aligned document sentence-aligned corpus, or any bilingual dictionary for this research analysis. For the most frequent word, we are assuming that there is an intra-lingual similarity distribution, and both the source and the target corpora have a comparable distribution graph. Additionally, these embeddings are isometric. These cross-lingual word embeddings are used for cross-lingual transfer learning and unsupervised neural machine translation. This research aims to improve the accuracy and efficiency of word translation between different language pairs by employing different retrieval techniques. The study analyzes the effectiveness of these techniques on different language pairs, including English-Hindi, English-Punjabi, English-Gujarati, English-Bengali, and English-Marathi. The research is expected to contribute significantly to the field of language translation by introducing innovative methods and other applications.

Keywords: cross-lingual embedding; word embedding; retrieval techniques; unsupervised word translation

ARTICLE INFO

Received: 21 November 2023
Accepted: 28 December 2023
Available online: 4 March 2024

1. Introduction

The explosion of textual data available on the Internet and the requirement to process and understand this vast amount of information have spurred significant advancements in Natural Language Processing (NLP). Large datasets and recent advances in deep learning models have led to various research studies in the NLP field, including Text Classification, Sentiment Analysis, and Information Retrieval (IR). Word embedding is a representation learning technique that emerged as a fundamental component in developing cutting-edge NLP models. It allows similar words to have similar vector representations, capturing syntactic and semantic information. Different types of research have been conducted in this area in various languages, but very few works are available in Indian languages. So, Word embeddings are particularly useful for Indian languages, due to the need for semantic representation, limited

labeled data, and the potential for cross-lingual applications. These techniques enable the development of robust NLP systems for Indian languages, improving their performance and accessibility.

Word embeddings have revolutionized the field of NLP by enabling machines to grasp the semantic relationships between words. For instance, using word embeddings, we can understand that "king" is related to "queen" in the same way "man" is related to "woman." The historical development of word embeddings involves early methods like one-hot encoding and count-based vectorization, which were limited in capturing word meanings. There are numerous ways to express language words in modern times. Word embeddings, a widely used word representation technique that connects a machine's language knowledge to a human's, is essential for resolving many NLP issues. A popular technique for learning word representation is word embedding, in which words with similar meanings have similar representations[1,2]. Some machine learning tasks are being aided by some conventional techniques, namely one-hot encoding and a bag of words, but because they are unordered, the context (surrounding words) and the frequency of words are lost.

However, according to Premjith et al.[3], these approaches do not provide any information regarding the semantics or the structural links between words. In word embedding, each real-valued number that makes up the representation of a word in n-dimensional space represents a different dimension of the meanings of the word. So as a result, the close vectors are associated with close words, and vice versa.

Google researchers have proposed a technique for learning word embeddings that are based on skip-gram or continuous bag-of-words architectures, both of these are implemented in Word2vec[1] and Fast-Text[4] libraries. FastText model that represents the sentences with a bag of n-grams, sub-word information, and shared vector data between the classes via a hidden representation. Researchers at Stanford University have proposed another approach, i.e., Glove, which maps words into a latent space in which the distance between words is correlated with semantic similarity[5].

Monolingual word vectors that independently trained for every language on its corpus in several NLP applications, particularly in Neural Machine Translation (NMT)[6]. The common space on a bilingual dictionary is represented by these monolingual vectors[7,8]. The cross-lingual word embedding model that permits the cross-lingual information transfer, is the mapping between word vectors.

Many studies on supervised and semi-supervised Machine Translation (MT) have been conducted recently by researchers. But nowadays, unsupervised MT is trending in research. Cross-lingual word embeddings are mainly used to translate the knowledge from one language to another language (i.e., source and target language) and vice-versa. Cross-lingual word representations provide a sophisticated and language-pair-independent method of representing text data in several languages[9].

The majority of these methods are supervised and make use of a large vocabulary made up of a few thousand pairs in order to learn the mapping in embedding space. The supervised method which generates the cross-lingual embeddings relies on thousands of parallel sentences. As a way to improve the performance of monolingual models in tasks requiring cross-lingual generalization, other modifications such as including noise, adapters, and language-specific position embeddings will be investigated[10]. One of the main ways that cross-lingual transfer is facilitated while constructing NLP models is through the representation of cross-lingual words, which allows us to reason about the meanings of words in multilingual contexts. Because cross-lingual embeddings share a common space, many machine translation tasks can make use of it[11]. Additionally, it can be applied to enhance language models[12]. Facebook AI has created the unsupervised MUSE[13] model, which is built on adversarial training in a favorable environment. The word-to-word findings for morphologically rich languages like Hindi are inadequate.

This paper's main contributions are:

- To explore the Cross-Lingual Word Embedding (CLWE) for different language pairs by unsupervised technique.

- To apply an unsupervised mapping technique for the different language pairs bilingual corpus.
- Using different word retrieval techniques such as cross-lingual word scaling, inverted softmax, and getting at least five nearest neighbors of the source word to target space.

In this paper, various word retrieval methods are carried out, and compare the embeddings of both languages; English and other Indian languages, that are trained for semi-supervised and unsupervised methods by passing a seed dictionary. An English and other Indian languages dictionary that has been produced is used to test bilingual word embedding. Several word retrieval techniques are available, and they are compared based on different factors such as training method, architecture, and performance. We will explore how researchers might apply cutting-edge techniques for building resource-light cross-lingual word representations in numerous downstream NLP applications.

The rest of this research paper is arranged as follows. Section 2 discusses the related work and Section 3 explains the word embedding, and different Word2Vec techniques and presents the benefits and limitations of these techniques. Then, Section 4 presents a cross-lingual word embedding overview, and word retrieval methods from cross-lingual embedding are explained. In Section 5 experimental settings are discussed and presents the results and discussions. Finally, concludes this paper and future work in the last section.

2. Literature review

Cross-lingual word embeddings represent continuous words in real vectors (or real numbers) in a shared vector space among various languages. This aids in determining potential word translations by calculating the distance between word embeddings across several languages. These embeddings are created by individually training word embeddings from two different languages, and they are then mapped to a common vector space via a linear transformation. Some of the systems are performing well for English to Hindi, the system that performs well for one language pair, but do not perform well for other languages.

There are some approaches that can be used for generating these embeddings categorized into regression methods, which can map embeddings of one language by considering a least squares objective[14,15]. Canonical methods map the word embeddings of both languages (i.e., source and target language) using canonical correlation analysis to a shared space. A bootstrapping method is used for a semi-supervised scenario to train the seed dictionary to have a few numbers of words. Low-resource languages that have less amount of corpora, unsupervised translation is more suitable for them. Similar research work was already investigated by Vulić and Korhonen[16] for count-based vector space models.

Recently, a self-learning concept by Artetxe et al.[17] that is initialized with just 25-word dictionary pairs and iteratively improves mapping through dictionary induction steps has been proposed. The results are approximations of other supervised approaches. Some other methods reduce the need for bilingual supervision by framing heuristics to have the seed dictionary. The cross-lingual embedding mappings using seed lexicon are studied by Smith et al.[18] an aligned document corpus is used to extract the training dictionary. These methods have a strong foundation in writing language systems even though they aim to do away with the need for bilingual data in use. A recent line of fully unsupervised techniques proposed by Artetxe et al.[17] where an encoder maps the one language (Source) embeddings into another (target), a decoder restores the one language (source) embeddings from the mapped embeddings, and a discriminator distinguishes between the mapped embeddings and the accurate another language (target) embeddings. Zhang et al.[19] use the same architecture but integrate additional techniques like noise addition to aid training and outline impressive results on bilingual lexicon translation. Zhang et al.[20] take on the earth mover's distance for training purposes and enhance it using a Wasserstein Generative Adversarial Network (GAN) ensue by an alternating optimization procedure. Several cross-lingual tasks, including sentiment analysis and machine translation, are made easier by cross-lingual word embedding. Low-resource languages have a little

amount of corpora so that unsupervised word translation is well suited for these languages. The intriguing new research on different cross-lingual word representations like supervised, semi-supervised, and unsupervised is thoroughly reviewed in this paper.

3. Word embedding models

Word embeddings is an NLP technique that is very useful for mapping words phrases or sentences in a high-dimensional vector space where the distance between two vector points represents their semantic similarity. There are various models that are used to generate word embeddings, including the most popular algorithms such as Word2Vec and GloVe[21]. Word embeddings for Indian languages have been constructed by using different techniques, creating multiple embeddings for these languages.

This review of the literature seeks to give an overview of the existing research on word embedding methods, exploring their theoretical foundations, methodologies, evaluation metrics, and applications. By examining the current state of the field, this review seeks to identify the strengths, limitations, and potential future directions for word embedding research. There are various techniques for generating word embeddings, some of the most popular techniques are:

3.1. Word2Vec

Word2Vec model is a neural network-based model that generates word embeddings by training on large text corpora. So, there are mainly two types of Word2Vec models: CBOW and Skip-gram.

3.1.1. CBOW

It is used to identify a word based on its context, that is the surrounding words. The model learns to predict the most probable target word given a context[22] (a set of surrounding words). It is a simple and faster model that discovers the most frequent words.

3.1.2. Skip-gram

It is used to identify the context of a given target word. The model learns to identify the most probable surrounding words (context) given a target word. To acquire the knowledge of monolingual source and target embeddings separately, run skip-gram[23] augmented with the character n-gram. Skip-gram is useful for representing rare words with very few datasets.

3.2. FastText

FastText[24,25] is an approach for computing word embeddings and it is an extension of Word2Vec models. This approach is CBOW and Skip-gram based that is used for representing each and every word as a bag of n-gram characters. Both of these Word2Vec models avoid the morphology of words. It uses the sub-word information to generate word embeddings, that are very effective for rare words and misspelled words. It generates word embeddings by considering character n-grams instead of whole words. It takes very less time to train on high-quality corpora and is very helpful in handling out-of-vocabulary (OOV) words and capturing sub-word information.

3.3. GloVe

GloVe is Globel Vectors for Word Representation which extends the Word2Vec model that learns the word vectors efficiently. Both Word2Vec and GloVe carry out the same tasks and in the NLP tasks, they also perform similarly. The way they are constructed is the major difference. A count-based model is GloVe whereas Word2Vec generates word embeddings by using a predictive model. GloVe generates word embeddings using co-occurrence information of words in a corpus. GloVe[24] uses a co-occurrence matrix to create word embeddings. By using matrix factorization, this model helps us to learn or discover the word representations.

To prepare for word embedding, one of the most important and useful techniques is Word2Vec. Word2Vec is a prediction-based technique that uses neural networks to generate word embeddings. Word2Vec is a type of neural network model used for NLP that produces word embeddings by training on high-quality text corpora. There have been several successful applications of word embeddings in MT for Indian languages, like English to other languages, and vice-versa. FastText[4] and GloVE[5] further improved on results, where FastText utilized the sub-word information to generate word vectors and GloVE used a co-occurrence matrix.

A fully unsupervised method[26–29] where an encoder maps the source language embeddings into the target language and from the mapped embeddings, a decoder extracts the source language embeddings, and then a discriminator separates the mapped embeddings from accurate target language embeddings. The quality of the models varied due to the differences in these language properties and corpus sizes. Word embedding is one of the major key technologies that are used to develop or improve more accurate and efficient machine translation systems for Indian languages.

4. Cross-Lingual word embedding (CLWE)

Cross-Lingual Word Embeddings (CLWE) are word representations in different languages that are mapped to a common space, enabling the exchange of meaning and knowledge across languages. The goal of CLWE is the alignment of the word embeddings that have been trained on both source and target corpora in a shared n-dimensional embedding space.

The existence of the geometrical symmetry between the word arrangement in two different languages, or between the embedding spaces of both (source and target) languages, was initially proposed by Dinu et al.[14]. With some linear transformation, one can learn mapping the source embedding space into the target embedding space. We attempt to reduce the distance between word pairs provided as a dictionary to model by continuously rotating the embedding. CLWE offers an elegant and language-pair-independent method for representing words across different languages[9].

In contrast with monolingual word embeddings, Cross-lingual word embeddings generate a shared projection between two monolingual vector spaces. MUSE[13] was implemented to get cross-lingual word embeddings across different languages. VecMap[17] implemented unsupervised learning for these word embeddings.

In semi-supervised learning for word translation, 25 words are used as a seed dictionary. And this seed dictionary is required to prevent inadequate local optimization. Unsupervised learning with the similarity distribution without using a seed dictionary creates a cross-lingual mapping of the word. In order to learn the linear transform, word embeddings must first be normalized, and a seed dictionary must then be initialized using the similarity distribution of the most similar words. A probability based on a robust self-learning technique is employed to optimize them[17]. The rotation matrix W, which has $N \times N$ dimensions and N depends on the size of the word embedding, is learned using adversarial learning. Its main purpose is to roughly align the source and target embedding. The generator makes an effort to deceive the discriminator such that it is unable to determine the origin of the embedding from either the source or the target. This process is trained to Matrix W, which then learns the weights to map the source language to the destination language embeddings[19].

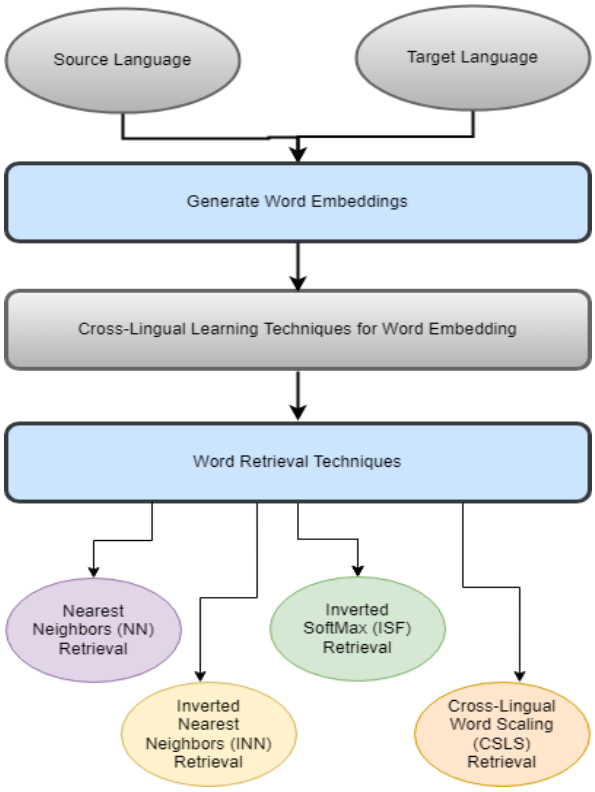


Figure 1. Cross-lingual word translation using different word retrieval techniques.

In **Figure 1**, the source and target languages' monolingual word embeddings are generated using the similarity distributions based on the most common words with the aim of aligning the word embeddings of both language pairs. And then getting the closely aligned embeddings for both languages (source and target) by using unsupervised learning technique.

To retrieve word-to-word translations from cross-lingual embeddings, we can use several word retrieval methods. Numerous retrieval methods are used to translate the embeddings from one language to another (Source and target language). Some of the popular retrieval techniques such as a nearest neighbor (NN), inverted nearest neighbor (INN), inverted softmax (ISF), and cross-lingual word scaling (CSLS) are discussed as follows.

4.1. Nearest neighbor (NN)

Nearest Neighbor or NN is a simple method used for pattern classification. In the context of cross-lingual word retrieval, it can be used to discover the most similar words in the target language (i.e., Hindi, Punjabi, Gujarati, Marathi, and Bengali) for a given word in the source language (i.e., English). The idea is to identify the k nearest neighbors in the target language for a given word in the source language. The words with the minimum distance to the given word are considered as the nearest neighbors.

The *NN* retrieval method is similarity matrix-based, often cosine similarity. In this method, we have taken the nearest embeddings in the target language. The *NN* of source embedding in the target space has maximum cosine similarity. The essence of the *NNs* is asymmetry: *y* being a *K-NN* of *x* does not imply that *x* is a *K-NN* of *y*. The position of *y* in the sorted list of similarities is represented by Equation (1).

$$NN = [cos(x, y_i)|y_i \in T] \quad (1)$$

$$NN_k(x, T) = \arg\frac{min}{y \epsilon T} Rank_{x,T}(y) \quad (2)$$

In Equation (2), *x* represents the source word vector, *y* represents the target neighbor, and *T* shows the target embedding space. $Rank_{x,T}(y)$ the rank of an element $y \in T$. The set of k nearest neighbors in *T* is represented by $NN_k(x, T)$. The *NN* of the *x* source word is represented by Equation (2).

Sometimes this retrieval technique does not properly work, because a few points are NNs of some other points, these are called as dubbed hubs. Some points are not nearest neighbors of any other points, called anti-hubs. The hubness problem, which refers to this issue, typically arises in higher-dimensional space. The inverse nearest neighbor retrieval (INN) retrieval method is proposed to tackle the hubness problem[16]. We took the target language's embedding and ranked it according to source embeddings in its neighbors to find the highest rank of the embedding in a target language.

For example, we have taken the word "happy" from the English language. By using the NN algorithm, we would find the Hindi word that has the most similar meaning to "happy". This could be the word "खुश" (Khush).

4.2. Inverted nearest neighbor (INN)

In cross-lingual embeddings, hubness problem is very common. A variation of the Nearest Neighbor (NN) retrieval method is the Inverted Nearest Neighbor (INN) used to overcome the hubness problem that occurs in the NN method. Instead of finding the nearest neighbors (NNs) of a given word in the target language, that can be Hindi, Punjabi, Guajarati, Marathi, or Bengali, INS finds the nearest neighbors of a given word in the source language, i.e., English. If the source language is more expressive than the target language, this approach is particularly helpful.

It uses the target language's embeddings and then ranks it based on the source embeddings in its neighbors. Then determines the highest rank embedding in a target language. Equation (3) employs the following globally corrected (GC) strategy, which may be easily implemented as follows:

$$GC(x,T) = \arg\tfrac{min}{y\epsilon T} Rank_{y,P}(x) \tag{3}$$

Here, returning the *NN* of pivot *x* serves as the source word vector, and *T* serves as the target embeddings. Then Equation (3), becomes the conventional *NN*, when only there is only one source vector. If the Rank is greater or equal to one, and the cosine is less than one. And then Equation (4) is implemented globally corrected (GC) as follows:

$$GC(x,T) = \arg\tfrac{min}{y\epsilon T} (Rank_{y,P}(x) - \cos(x,y)) \tag{4}$$

For example, we have taken the word "खुश" (Khush) from the Hindi language. If we were to use the Inverted NN algorithm, we would find the English word that has the most similar meaning to "खुश". This could be the word "happy" in English.

4.3. Inverted SoftMax (ISF)

Inverted SoftMax (ISF) is a technique used in machine learning for finding the most likely class of a given sample. In the context of cross-lingual word retrieval, it is used to determine the most likely translation of a word from one language to another (i.e., source and target). This method involves calculating the SoftMax of the dot product between the embeddings of the word in the source language and the embeddings of all words in the target language.

To reduce the hubness problem which is similar to the inverted nearest neighbor (INN) retrieval method, the inverted SoftMax retrieval is used[16]. ISF retrieval technique uses the SoftMax function with a few hyper-parameters, instead of using the cosine similarity for calculation. And over the source words, the probability normalization is used by this retrieval method. It also works by reversing the query direction, but when computing similarity, it employs a SoftMax function rather than the cosine as follows:

$$P_{j\to i} = \frac{e^{\beta S_{ij}}}{\alpha_j \sum_n e^{\beta S_{in}}} \tag{5}$$

Equation (5), determines the translation of the source word *j*th by locating the target *i*th word. Choose the target word, that has the highest probability. The denominator in Equation (5), will be large if the *i*th target word is a hub, which will prevent the target word from being chosen. Vector α ensures normalization.

For example, we have taken the word "happy" from the English language. If we were to use the Inverted SoftMax technique, we would identify the Hindi word that has the highest probability of being the correct translation for "happy". This could be the word "खुश" (Khush) in Hindi.

4.4. Cross-Lingual word scaling (CSLS)

Cross-lingual word Scaling is a retrieval method used to align word embeddings across different languages. It is a method for enhancing the performance of NN algorithms in cross-lingual word retrieval. It involves scaling the word embedding of a source language so that they have the same average as the word embedding of a target language. This scaling is done before the NN algorithm is applied, and it helps to reduce the distance between the embeddings of words that have the same meaning but are expressed differently in both source and target languages (i.e., English and Hindi).

$$r_T(Wx_s) = \frac{1}{k} \sum_{y_1 \epsilon N_T(Wx_s)} \cos(Wx_s, y_T) \tag{6}$$

$$CSLS(Ux_s; y_t) = 2\cos(Ux_s, y_t) - r_T(Ux_s) - r_s(y_t) \tag{7}$$

A Cross-lingual word scaling's similarity measure between the source and target words mapping is represented in Equation (7), where, $r_s(y_t)$ is the mean similarity of y_t which is a target word. These values are then calculated for all word vectors (source and target) with efficient nearest neighbors.

For example, we have the term "happy" in English. If we were to employ the CSLS method, we would scale the word embedding of "happy" in order for it to have the same average as the word embeddings of words that have the same meaning in Hindi. This would help in reducing the distance between the embeddings of "happy" and its Hindi translations, making it simpler for the NN retrieval method to identify the most similar words.

5. Experimental setup

5.1. Dataset

For this research study, we used bilingual corpus test datasets for different languages, i.e., English, Hindi, Punjabi, Gujarati, Marathi, and Bengali. For generating cross-lingual word embedding for these language pairs, we have taken a monolingual corpus of AI4Bharat[30], IIT Bombay[31], and WMT[32] datasets.

5.2. Data pre-processing

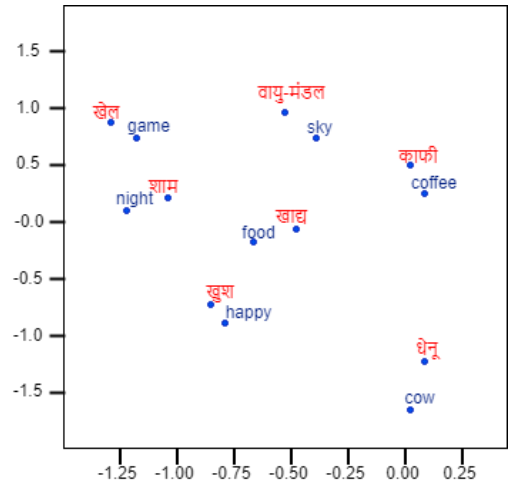
In any machine translation process, the data pre-processing is very important. The corpus directly influences the quality of the embedding. We utilized common technologies from the natural language toolkit for the English and other language corpus.

5.3. Training phase

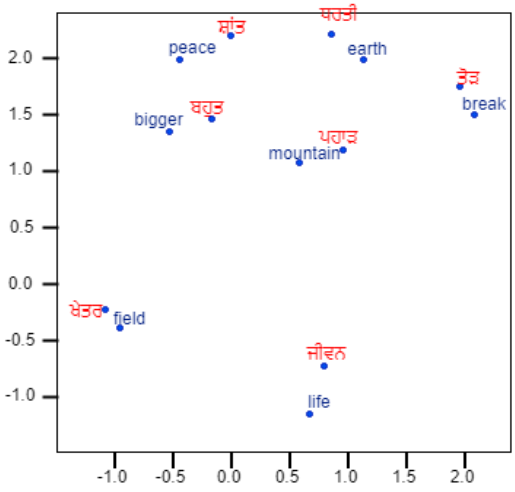
For this research work, we used the FastText word embedding model. FastText was developed by Facebook's AI Research lab. It is a library for effective word representation and sentence classification learning. This approach is useful for the tasks that involve text data, like information retrieval, text classification, and spell-checking. The key benefits of this approach are speed and efficiency, which make it appropriate for huge datasets. FastText with a skip-gram model is mainly used for generating the word vectors. The FastText approach deals with word morphological changes and out-of-vocabulary issues. It also supports the sub-word information and is also useful for languages with complex morphology.

We generated the mapped word embeddings of source and target, that have the most frequent words. Several word retrieval techniques are used for extracting the dictionaries. It is necessary to select the word embedding of the source language that has the highest word similarity with the word embedding of the target language.

Some of the Hindi and English language words are plotted in the mapped area to demonstrate mapping in a two-dimensional shared space using the nearest neighbor concept as shown in **Figure 2**. Principal Component Analysis (PCA) is used to minimize the dimensionality of a sample of randomly selected words from various domains. Some terms, like "night" and "शाम" which are used in close proximity to one another, are easily translated from one to the other as shown in **Figure 2a** and in **Figure 2b** the word "earth" and "ਧਰਤੀ" which are used in close proximity to one another, translated from one to the other. The languages Hindi and Punjabi are closely related and follow the same word order.



(a) English and Hindi.



(b) English and Punjabi.

Figure 2. A shared embedding space for both languages.

6. Results and analysis

6.1. Nearest neighbors for Hindi as the target language

We are predicting the nearest top five scores for source words of English in the target space of Hindi. The load_vec function reads word embeddings from a file. It returns three values: A numpy array of embeddings, a dictionary mapping words to their IDs, and a dictionary mapping IDs to words. The function stops reading after nmax words. The get_nn function computes the cosine similarity between the embeddings

of a given word in the source language and all words in the target language. The cosine similarity is calculated as the dot product of the normalized embeddings. It then prints the K words in the target language with the highest cosine similarity. Here, Sr. No. indicates the Serial Number, and we have given the source words world, farmer, night, category, and government, and then determined the top five closest Hindi words for the given source words as shown in **Table 1**. It has been determined that a word's prediction directly depends upon the word's frequency in the corpus, with a higher frequency of words being translated more accurately.

Table 1. Nearest neighbor of the English-Hindi language word translation.

Source word	**World**		**Farmer**		**Night**		**Category**		**Government**	
Sr. No.	**Similarity score**	**Word**	**Similarity score**	**Word**	**Similarity score**	**Word**	**Similarity score**	**Word**	**Similarity score**	**Word**
1	0.6465	दुनिया	0.7281	किसान	0.6148	रात	0.7283	वर्ग	0.7824	सरकार
2	0.6413	संसार	0.6296	कृषक	0.5962	रात्रि	0.6928	श्रेणी	0.6473	शासन
3	0.6205	लोक	0.5073	क्षेत्रजीवी	0.5725	रात्रि	0.6376	कोटि	0.6203	राज्य
4	0.5629	संसार	0.4765	क्षेत्रजीवी	0.5392	शाम	0.6302	पद	0.6143	सरकार
5	0.5437	विश्व	0.4298	खेतिहर	0.5374	रात्रि-चर	0.5634	श्रेणी	0.5339	सत्ता

Cosine similarity is the similarity metric between the two non-zero vectors of an inner product space. It is used in Word2Vec to find the most similar words to a given word. **Table 2** shows the results of a word similarity analysis using cosine similarity for the given word "like" in the source language, i.e., English using different retrieval techniques.

Based on their cosine similarity of the word "like", the words are ranked as shown in **Table 1**. In this table, NN represents Nearest Neighbour, INN as Inverted Nearest Neighbour, ISF as Inverted SoftMax, and CSLS as Cross-Lingual Word Scaling. When the cosine similarity is highest, then the more similar the word is to "like". For instance, the word "likes" has a cosine similarity of 0.8109, indicating that it is more similar to "like" than the other words in **Table 2**. Hindi as a source language, the words are ranked based on their cosine similarity to the word "बड़ा". **Table 3** presents the Cosine similarity to the given word "बड़ा" in the source language, i.e., Hindi using different retrieval techniques**.** The cosine similarity is higher, then the more similar the word is to "बड़ा". For example, the word "बड़बड़ाहट" has a cosine similarity of 0.9441, indicating that it is more similar to "बड़ा" than the other words as shown in **Table 3**.

Cosine similarity calculates the cosine of the angle between two vectors. It is a measurement of orientation and not magnitude, it can be thought of as a comparison between two documents on a normalized space. This is a tool that is used to measure the similarity between two vectors.

Table 2. Word similarity for the Source word: Like.

SOURCE_WORD: Like								
Retrieval technique	**NN**		**INN**		**ISF**		**CSLS**	
Sr. No.	**Cosine similarity**	**Word**	**Cosine similarity**	**Word**	**Cosine similarity**	**Word**	**Cosine similarity**	**Word**
1	1.0000	Like	−0.5730	bedroom	0.0006	like	0.0918	like
2	0.8109	likes	−0.5178	statue	0.0004	make	0.0911	liked
3	0.7702	Iike	−0.4893	Room	0.0003	That	0.0897	likelihood
4	0.7317	likely	−0.4876	room	0.0002	Then	0.0864	like
5	0.7206	Like	−0.4860	fire	0.0002	some	0.0791	likes

Table 3. Word similarity for the Source word: बड़ा.

SOURCE_WORD: बड़ा								
Retrieval technique	**NN**		**INN**		**ISF**		**CSLS**	
Sr. No.	**Cosine similarity**	**Word**	**Cosine similarity**	**Word**	**Cosine similarity**	**Word**	**Cosine Similarity**	**Word**
1	1.0000	बड़ा	−0.0054	प्रतिबद्धता	0.0002	प्रभावी	0.0738	बड़ा
2	0.9441	बड़बड़ाहट	0.0013	प्रशासन	0.0002	अद्भुत	0.0715	कुबड़ा
3	0.9077	कुबड़ा	0.0100	लोकतंत्र	0.0002	अलंकार	0.0692	प्रभावी
4	0.8240	चीथड़ा	0.0108	चुना	0.0001	प्रबल	0.0641	बड़बड़ाना
5	0.8048	लँगड़ा	0.0124	प्रतिबंध	0.0001	लंबे	0.0579	बड़ा

6.2. Accuracy (%) calculation for English to other languages and vice-versa

To calculate the accuracy of different Indo-Aryan language pairs, we provided the bi-lingual English language to other different target language dictionaries. English source words have multiple target language translations. **Table 4** presents English as a source language to five target languages using four retrieval techniques and evaluates the accuracy for each language pair.

Table 4. Accuracy (%) between English (i.e., source language) to other languages.

Retrieval techniques	**English as a Source language to other Target language Pair**				
	Eng-Hi	**Eng-Pbi**	**Eng-Guj**	**Eng-Ben**	**Eng-Ma**
Nearest Neighbour	48.89	42.31	34.24	33.54	32.18
Inverted Nearest Neighbour	48.27	39.52	36.41	29.63	27.16
Inverted SoftMax	37.64	34.83	30.12	32.47	22.53
Cross-Lingual Word Scaling	49.75	43.96	39.27	33.78	32.64

An accuracy of 48.89% is obtained when the nearest neighbour retrieval technique is used for the English-Hindi language pair. It provides the highest accuracy as compared to the other language pairs using the nearest neighbour technique and English- Marathi language pair has the 32.18% which is the lowest accuracy. But when the Cross-Lingual Word Scaling (CSLS) retrieval is used, the accuracy of 49.75% is obtained for English-Hindi language pair. It clearly shows that the CSLS performs well because by removing the hubness problem that occurred in the Nearest neighbor retrieval technique. The nearest neighbour obtained the 42.31%, inverted nearest neighbour 39.27%, inverted SoftMax 34.83%, and Cross-lingual word scaling gives highest accuracy 43.96% for English-Punjabi language pair. For English-Gujarati language pair, Cross-lingual word scaling provides highest accuracy 39.27% and Inverted SoftMax gives lowest accuracy, i.e., 30.12%. For English-Bengali language pair, inverted nearest neighbour has lowest accuracy 29.63%, and Cross-lingual word scaling has 33.78% highest accuracy. For English-Marathi language pair, Cross-lingual word scaling provides highest accuracy 32.64% and Inverted SoftMax has 22.53% lowest accuracy.

Figure 3 shows the accuracy in percentage for different language pairs using four retrieval techniques. In this figure, English is represented as Eng, Hindi as Hi, Punjabi as Pbi, Gujarati as Guj, Bengali as Ben, and Marathi as Ma. And this graph shows that Eng-Hi language pair has the maximum accuracy as compared to the other language pairs. Cross-lingual word scaling is the best retrieval technique among the other four retrieval techniques.

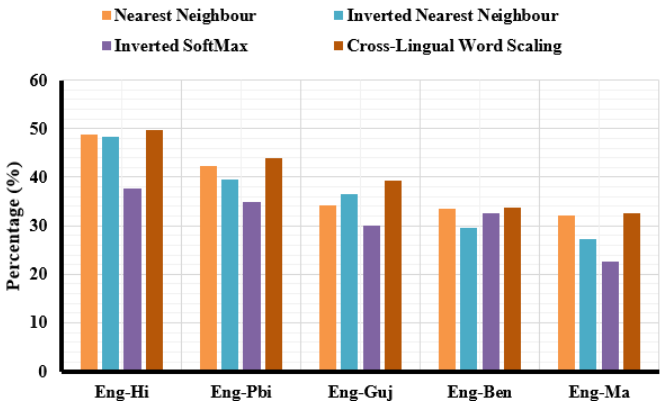


Figure 3. Accuracy (%) calculation for different language pairs using different retrieval techniques (English as a source language).

Table 5 presents the English as a target language to different source languages using four retrieval techniques and evaluates the accuracy for each language pair. Here, Hindi, Punjabi, Gujarati, Bengali, and Marathi are the source languages and English is a target language.

Table 5. Accuracy (%) between different Source languages to English language.

Retrieval techniques	**Source languages to English as a Target language Pair**				
	Hi-Eng	**Pbi-Eng**	**Guj-Eng**	**Ben-Eng**	**Ma-Eng**
Nearest neighbour	47.68	41.38	34.12	32.46	30.76
Inverted nearest neighbour	46.72	38.25	34.59	28.31	27.13
Inverted SoftMax	36.74	33.81	29.13	31.47	22.83
Cross-Lingual word scaling	48.53	42.75	38.76	34.18	33.62

When the nearest neighbour retrieval technique is used for the Hindi-English language pair, an accuracy of 47.68% is obtained. It provides the highest accuracy as compared to the other language pairs using the nearest neighbour technique and the Marathi-English language pair has the 30.76% which is the lowest accuracy. But when the Cross-Lingual Word Scaling (CSLS) retrieval is used for the Hindi-English language pair, an accuracy of 48.53% is obtained. The nearest neighbour technique obtained the 41.38%, inverted nearest neighbour 38.25%, inverted SoftMax 33.81%, and Cross-lingual word scaling gives the highest accuracy 42.75% for the Punjabi-English language pair. For Gujarati-English language pair, Cross-lingual word scaling provides the highest accuracy 38.76% and Inverted SoftMax gives lowest accuracy, i.e., 29.13%. For the Bengali-English language pair, nearest neighbour has 32.46%, the inverted nearest neighbour has 28.31%, inverted SoftMax has 31.47%, and Cross-lingual word scaling has 34.18% accuracy. For the Marathi-English language pair, nearest neighbour has 30.76%, the inverted nearest neighbour has 27.13%, inverted SoftMax has 22.83%, and Cross-lingual word scaling has 33.62% accuracy.

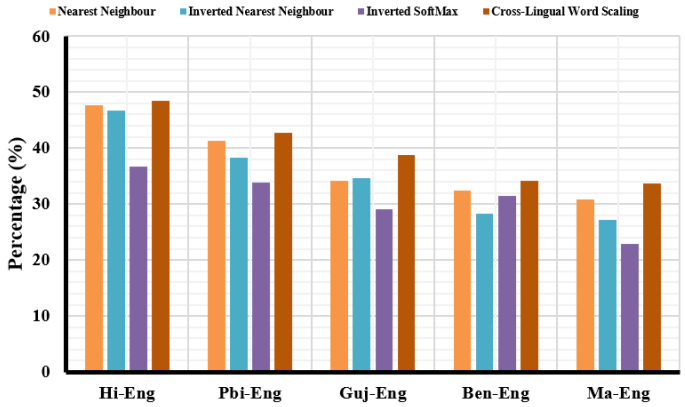


Figure 4. Accuracy (%) calculation for different language pairs using different retrieval techniques (English as a target language).

Figure 4 represents the accuracy in percentage (%) when English as a target language to five source languages using four retrieval techniques. This graphical representation provides the accuracy for each language pair in percentage using different retrieval techniques.

7. Conclusion and future scope

The supervised learning technique uses a lot of parallel data for language translation. However, it takes a lot of time and human involvement. A cross-lingual embedding and a dictionary are required for the semi-supervised technique. A few research studies have been conducted by the researchers on unsupervised cross-lingual word embedding. In this research, we try to overcome this issue and produce the most effective cross-lingual word embeddings for English as compared to other Indian language pairs. We have taken a bilingual corpus test dataset for different Indian languages. These methods include a nearest neighbor, inverted nearest neighbors, inverted SoftMax, and cross-lingual word scaling. All these language pairs dictionary that has been produced is used to test bi-lingual word embedding. Bi-lingual dictionary for these languages.

In this study, we carried out word translation tasks from English to other Indian languages and vice-versa. To translate words from English to other Indian languages with rich morphological patterns, we applied a variety of word retrieval approaches based on mapped cross-lingual embedding. In the future, word embedding for numerous languages in a single plane will be attempted as a cross-lingual work-generation method. For future work, we intend to consider other Indian languages from different language families. In order to contrast our findings with those of other supervised and semi-supervised learning techniques.

Author contributions

Conceptualization, KK and SC; methodology, SC; software, SC; validation, KK and SC; formal analysis, KK; investigation, KK and SC; resources, SC; data curation, KK; writing—original draft preparation, KK; writing—review and editing, KK; visualization, KK; supervision, SC; project administration, SC; funding acquisition, KK. All authors have read and agreed to the published version of the manuscript.

Conflict of interest

The authors declare no conflict of interest.

References

1. Andrabi SAB, Wahid A. Machine Translation System Using Deep Learning for English to Urdu. Gupta SK, ed. Computational Intelligence and Neuroscience. 2022, 2022: 1-11. doi: 10.1155/2022/7873012
2. Kim Y, Geng J, Ney H. Improving Unsupervised Word-by-Word Translation with Language Model and Denoising Autoencoder. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Published online 2018. doi: 10.18653/v1/d18-1101
3. Premjith B, Kumar MA, Soman KP. Neural Machine Translation System for English to Indian Language Translation Using MTIL Parallel Corpus. Journal of Intelligent Systems. 2019, 28(3): 387-398. doi: 10.1515/jisys-2019-2510
4. Bojanowski P, Grave E, Joulin A, et al. Enriching Word Vectors with Subword Information. Transactions of the Association for Computational Linguistics. 2017, 5: 135-146. doi: 10.1162/tacl_a_00051
5. Pennington J, Socher R, Manning C. Glove: Global Vectors for Word Representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Published online 2014. doi: 10.3115/v1/d14-1162
6. Joulin A, Grave E, Bojanowski P, et al. Bag of Tricks for Efficient Text Classification. Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers. Published online 2017. doi: 10.18653/v1/e17-2068

7. Artetxe M, Labaka G, Agirre E. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. Published online 2016. doi: 10.18653/v1/d16-1250
8. Artetxe M, Labaka G, Agirre E. Learning bilingual word embeddings with (almost) no bilingual data. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Published online 2017. doi: 10.18653/v1/p17-1042
9. Ruder S, Søgaard A, Vulić I. Unsupervised Cross-Lingual Representation Learning. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts. Published online 2019. doi: 10.18653/v1/p19-4007
10. Artetxe M, Ruder S, Yogatama D. On the cross-lingual transferability of monolingual representations. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Published online 2020: 4623–4637. doi: 10.18653/v1/2020.acl-main.421.
11. Zou WY, Socher R, Cer D, Manning CD. Bilingual word embeddings for phrase-based machine translation. In: Proceedings of the 2013 conference on empirical methods in natural language processing.
12. Adams O, Makarucha A, Neubig G, et al. Cross-lingual word embeddings for low-resource language modeling. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics.
13. Conneau A, Lample G, Ranzato MA, et al. Word translation without parallel data. arXiv preprint. arXiv:1710.04087.
14. Dinu G, Lazaridou A, Baroni M. Improving zero-shot learning by mitigating the hubness problem. arXiv preprint. arXiv:1412.6568.
15. Shigeto Y, Suzuki I, Hara K, et al. Ridge Regression, Hubness, and Zero-Shot Learning. Lecture Notes in Computer Science. Published online 2015: 135-151. doi: 10.1007/978-3-319-23528-8_9
16. Vulić I, Korhonen A. On the Role of Seed Lexicons in Learning Bilingual Word Embeddings. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Published online 2016. doi: 10.18653/v1/p16-1024
17. Artetxe M, Labaka G, Agirre E. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Published online 2018. doi: 10.18653/v1/p18-1073
18. Smith SL, Turban DH, Hamblin S, Hammerla NY. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. arXiv preprint. arXiv:1702.03859.
19. Zhang M, Liu Y, Luan H, Sun M. Adversarial training for unsupervised bilingual lexicon induction. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics.
20. Zhang M, Liu Y, Luan H, Sun M. Earth mover's distance minimization for unsupervised bilingual lexicon induction. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing.
21. Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. In: Advances in neural information processing systems.
22. Hendy A, Abdelrehim M, Sharaf A, et al. How good are gpt models at machine translation? A comprehensive evaluation. arXiv preprint. arXiv:2302.09210.
23. Rivera-Trigueros I. Machine translation systems and quality assessment: a systematic review. Language Resources and Evaluation. 2022, 56(2): 593-619.
24. Rei R, Guerreiro NM, Treviso M, et al. The inside story: Towards better understanding of machine translation neural evaluation metrics. arXiv preprint. arXiv:2305.11806.
25. Singh M, Kumar R, Chana I. Machine translation systems for Indian languages: review of modelling techniques, challenges, open issues and future research directions. Archives of Computational Methods in Engineering. 2021, 28, 2165-2193.
26. Ramesh A, Parthasarathy VB, Haque R, Way A. Comparing statistical and neural machine translation performance on hindi-to-tamil and english-to- tamil. Digital. 2021, 1(2): 86-102.
27. Garje GV, Bansode A, Gandhi S, et al. Marathi to English Sentence Translator for Simple Assertive and Interrogative Sentences. International Journal of Computer Applications. 2016, 138(5): 42-45. doi: 10.5120/ijca2016908837
28. Chauhan S, Saxena S, Daniel P. Fully unsupervised word translation from cross-lingual word embeddings especially for healthcare professionals. International Journal of System Assurance Engineering and Management. 2021, 13(S1): 28-37. doi: 10.1007/s13198-021-01182-z
29. Chauhan S, Shet JP, Beram SM, et al. Rule Based Fuzzy Computing Approach on Self-Supervised Sentiment Polarity Classification with Word Sense Disambiguation in Machine Translation for Hindi Language. ACM Transactions on Asian and Low-Resource Language Information Processing. 2023, 22(5): 1-21. doi: 10.1145/3574130
30. Kunchukuttan A, Kakwani D, Golla S, et al. Ai4bharat-indicnlp corpus: Monolingual corpora and word embeddings for indic languages. arXiv preprint. arXiv:2005.00085.
31. Kunchukuttan A, Mehta P, Bhattacharyya P. The it bombay English-Hindi parallel corpus. arXiv preprint. arXiv:1710.02855.

32. Post M, Callison-Burch C, Osborne M. Constructing parallel corpora for six Indian languages via crowdsourcing. In: Proceedings of the seventh workshop on statistical machine translation.